Robust Watermarking of Video Streams

T. Polyák

In the past few years there has been an explosion in the use of digital video data. Many people have personal computers at home, and with the help of the Internet users can easily share video files on their computer. This makes possible the unauthorized use of digital media, and without adequate protection systems the authors and distributors have no means to prevent it.
Digital watermarking techniques can help these systems to be more effective by embedding secret data right into the video stream. This makes minor changes in the frames of the video, but these changes are almost imperceptible to the human visual system. The embedded information can involve copyright data, access control etc. A robust watermark is resistant to various distortions of the video, so it cannot be removed without affecting the quality of the host medium. In this paper I propose a video watermarking scheme that fulfills the requirements of a robust watermark.

Keywords: Robust watermarking, video streaming.

1 About watermarking

Watermarking is a type of steganography, with the help of which we can embed information into the host medium, in our case into a video stream. Its main task is to protect the medium. To embed the information we use a key (stego key), and the same key is used during the watermark detection process [1].

The watermarking process consists of the following steps (Fig. 1.):

Creation of the watermark (W): coding the data to embed (D) into a format that is acceptable as input for the embedding algorithm.

Using the stego key (S), we embed the watermark into the host medium (M). The medium, which contains the watermark is called a stego medium (M_w).

The medium passes through a transmission channel, where it can suffer various distortions (intended or unintended). We will refer to this as a distorted medium (M_w').

Using the stego key again, we search the hidden watermark in the medium. The watermark has probably suffered distortions too, and we have to watch for this during retrieval. Some algorithms use the original medium, but this is not necessary. If we detect the watermark without the original medium, we speak about blind watermarking.

We decode the retrieved watermark, and get the embedded data.

2 Requirements of watermarking

If we want to design a watermarking system to protect our medium we must agree on numerous requirements [2].

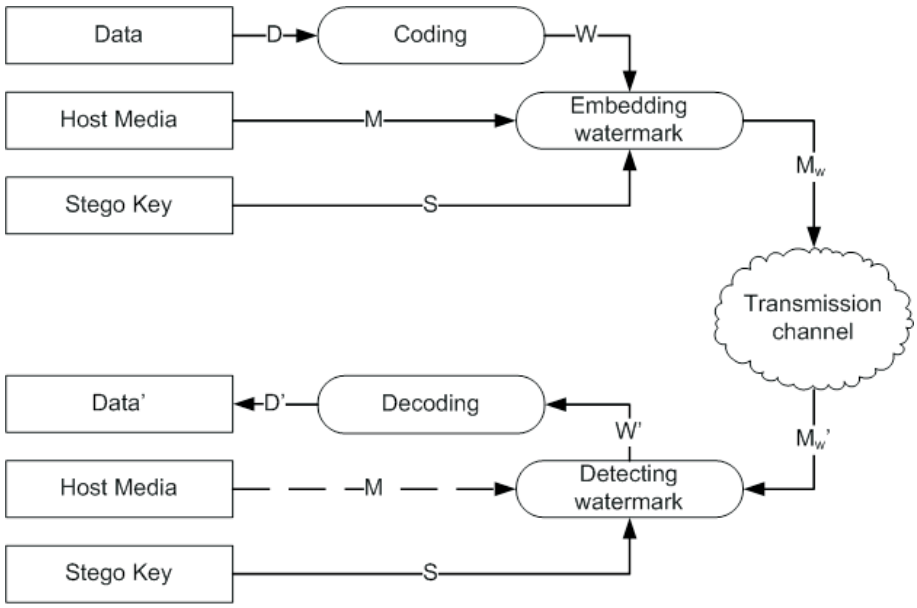

Fig. 1: Block scheme of watermarking

Robustness: this refers to the resistance of the watermark to various distortions and/or attacks. There are many types of attacks that can impact a watermark: clipping, filtering, noise adding, etc. The watermark should not be deleted without significant loss of quality.

Imperceptibility: the watermark cannot be seen. This means that it can only make changes that are almost invisible to the human visual system, and it should not affect the quality of the video very much.

Payload: different applications need watermarks of different sizes. For copy protection, one bit is usually enough, but if we want to store copyright information, we need more bits.

The requirements mentioned above are slightly conflicting: if we maximize robustness, it will worsen imperceptibility, if we maximize imperceptibility, we can only hide a few bits. So we have to find the correct relationship between them (Fig. 2).

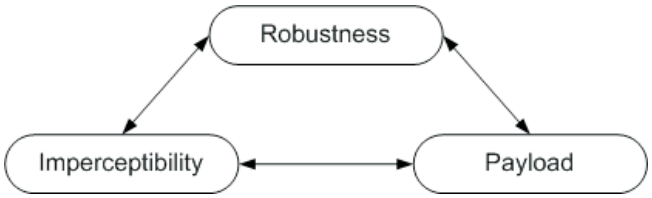

Fig. 2: Blockscheme of watermarking

3 Overview of the algorithm

The algorithm is designed to embed a watermark in small resolution (176×144 pixels) video files that can be played on mobile devices. These devices have limited computational capacities, so the algorithm should not be too complex.

The algorithm is based on the Dittmann algorithm, which is designed to watermark MPEG video files [3]. According to the MPEG DCT coding, Dittmann uses 8×8 pixel blocks to hide information. Our algorithm better fits more to the needs of mobile devices.

To improve robustness the algorithm uses larger blocks than Dittmann's algorithm. According to the key, different pattern blocks are added to or subtracted from the selected blocks of video frames. These pattern blocks are not counted in real time (to reduce the calculation requirements), they are read from a file.

The embedding algorithm obtains the following data as input. These are the free parameters of the algorithm.

The stego key.

The information to embed.

The strength of the watermark (n): this shows the amount by which the pixel luminance values are modified. Obviously, if we select a high value we improve robustness but worsen quality, and vice versa.

Pattern blocks: these are 0–1 blocks, and we hide the information with the help of these. First they are filled with random 0–1 values, then we eliminate the high frequencies. After this, there will be large 0 and 1 islands, so that during watermark detection the patterns can be more easily recognized, and the watermark will be more resistant to attack.

3.1 Embedding process

Each frame of the video is watermarked. When we embed the watermark in the frame, first we select the blocks of the frame which will contain the information bits (each block holds one bit of information) and the pattern blocks used for it. Information is hidden in the luminance values of the pixels. The method is as follows:

Counting the values of the pattern block: where the value of the chosen pattern is 1, we select n (n is the strength of the watermark), and where it is 0, the selected value is $-n$.

The scaled patterns are added to the frame blocks: if we want to hide 1 in the block, then we add the pattern to it. If we want to hide 0, we subtract it. The characteristics of the embedded watermark are shown in Fig. 3. There are eight watermarked blocks in the frame. High strength (n) is used for better visibility.

Fig. 3: Effect of the watermark on a grey frame

To improve robustness, the same data is hidden in five frame blocks. The algorithm also places synchronization information, so it can resist attacks in the time domain, e.g., frame dropping.

3.2 Detection process

During detection, the same blocks and pattern blocks are chosen as in the embedding process. Then the algorithm reads the bits of the blocks. It summarizes the pixel values according to the pattern (if the pattern value is 1, the pixel value is added to the sum, if it is 0, then it is subtracted from it). Due to the correlation of the bits, the sum should be around $2n$ or $-2n$. If the sum is positive, the hidden bit was 1, otherwise it was 0.

4 Test results

The tests of the watermark were of two main types: imperceptibility tests and robustness tests.

In the imperceptibility tests I studied the effect of watermarks of various strengths on the quality of the video. I put the watermark in the video stream, compared it to the original, and calculated the PSNR value caused by the watermark. Sample frames can be seen in Figs. 4–7, and the test results in Fig. 8.

In the robustness tests I submitted the watermarked videos to various attacks and distortions (cropping, filtering, noise adding, format conversions, etc.) For the strength of the watermarks I used values 1, 3, 5, 8 and 10. After attacking

Fig. 4: Original video

Fig. 5: Watermarked video (n=3)

Fig. 6: Watermarked video (n=5)

Fig. 7: Watermarked video (n=10)

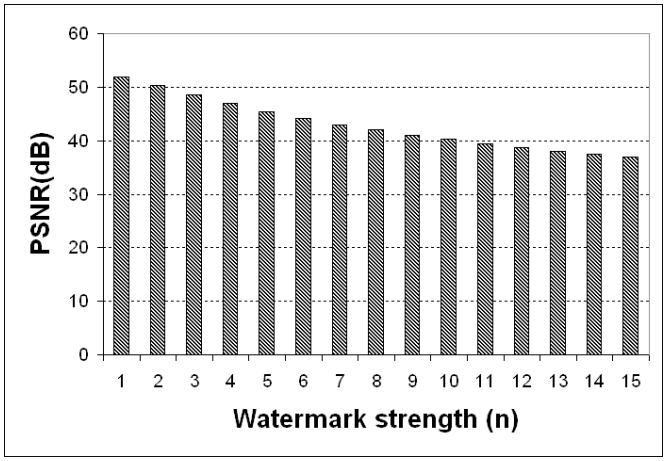

Fig. 8: PSNR values of video streams watermarked with different strength

the images I examined how many bits were read correctly by the detection algorithm. Some test results are shown in Fig. 9.

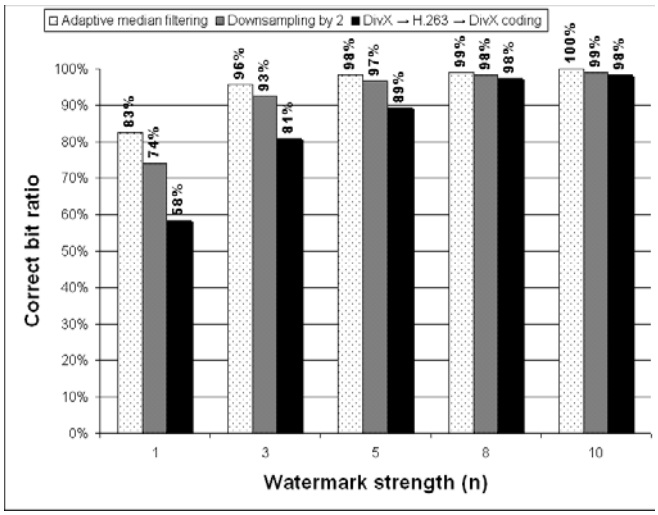

Fig. 9: Correct bit ratio after attacks (adaptive median filtering, downsampling, format conversion)

5 Summary

The test results show that there are appropriate settings of the algorithm which can ensure robustness of the watermark and low loss of the original quality of the video. The algorithm is quite simple, it does not use complex calculations, so it can be used in mobile devices.

References

[1] Furht, B., Kirovski, D.: *Multimedia Security Handbook*, CRC PRESS, 2005.

[2] Hartung, F., Kutter., M.: Multimedia watermarking techniques. *Proceeding IEEE*, Vol. **87** (1999), No. 7, p. 1079–1107.

[3] Dittmann, J., Stabenau, M., Steinmetz, R.: *Robust MPEG video watermarking technologies*, 1998.

Tamás Polyák
e-mail: polyak.tamas@tmit.bme.hu

Dept. of Telecommunication and Mediainformatics

Budapest University of Technology and Economics,
Műegyetem rkp. 3–9
Budapest, Hungary